Addressing uncertainty in atomistic machine learning

Peterson, Andrew A. ; Christensen, Rune; Khorshidi, Alireza

Published in:
Physical Chemistry Chemical Physics

Link to article, DOI:
10.1039/c7cp00375g

Publication date:
2017

Document Version
Peer reviewed version

Perspective:

Addressing uncertainty in atomistic machine learning

Andrew A. Peterson[*,1], Rune Christensen[2], Alireza Khorshidi[1]

[1]*School of Engineering, Brown University, Providence, Rhode Island, 02912, United States.*
[2]*Department of Energy Conversion and Storage, Technical University of Denmark, Kgs. Lyngby DK-2000, Denmark.*
**andrew_peterson@brown.edu*

Machine-learning regression has been demonstrated to precisely emulate the potential energy and forces that are output from more expensive electronic-structure calculations. However, to predict *new* regions of the potential energy surface, an assessment must be made of the credibility of the predictions. In this perspective, we address the types of errors that might arise in atomistic machine learning, the unique aspects of atomistic simulations that make machine-learning challenging, and highlight how uncertainty analysis can be used to assess the validity of machine-learning predictions. We suggest this will allow researchers to more fully use machine learning for the routine acceleration of large, high-accuracy, or extended-time simulations. In our demonstrations, we use a bootstrap ensemble of neural network-based calculators, and show that the width of the ensemble can provide an estimate of the uncertainty when the width is comparable to that in the training data. Intriguingly, we also show that the uncertainty can be localized to specific atoms in the simulation, which may offer hints for the generation of training data to strategically improve the machine-learned representation.

Atomic-scale calculations based on quantum mechanics are revolutionizing our understanding of and ability to design new materials, reactions, and devices. However, these calculations are computationally expensive, and there is a trade-off between the accuracy of the calculation and the computational requirements. The most accurate methods are typically limited to few atoms, and even popular methods like density functional theory (DFT) quickly become limited in size, since the computational requirements typically scale with the number of electrons (or electronic basis functions) cubed.[1] In many cases, the desired output of such electronic-structure calculations is simply the ground-state potential energy of the system, which is referred to as the (ground-state) potential energy surface (PES) when expressed as a function of atomic coordinates, $E(\vec{\mathbf{R}})$. In other cases, the atomic forces are desired, which can be derived from the PES as the (negative) gradients of the potential energy. Researchers typically perform many calculations that differ only in the geometric positions ($\vec{\mathbf{R}}$) or number (N) of atoms, and which therefore contain much redundant information. In recent years, several groups have suggested that machine-learning regression can be used to fit the PES, and thus can emulate the output of electronic-structure calculations.[2–7] This in principle can be done to arbitrary accuracy, given enough relevant training data and a flexible

enough regression model.[7,8] However, the question of how much and what kind of training data is necessary to make accurate predictions is challenging, and will be addressed in this work.

An in-depth discussion of the representation of the potential energy surface with machine learning can be found in our previous work[7] or other reviews;[9–11] here, we give only a brief background on the mathematical forms typically used in atomistic machine learning. In an atom-centered model, the energy and forces are predicted individually per atom. In this mode, a feature vector is constructed that describes the local environment around each atom; this provides desirable properties like size extensibility and invariance to permutation, translation, and rotation. Mathematically, this takes a form like

$$E^{\text{pred}} = \sum_{i=1}^{N_{\text{atoms}}} \hat{E}_i \left(\mathbf{G}_i \left(\vec{\mathbf{R}} \right) \right). \tag{1}$$

That is, a feature vector $\mathbf{G}_i(\vec{\mathbf{R}})$ is constructed for each atom (indexed by i) based on the coordinates $\vec{\mathbf{R}}$ of all the atoms in the image, although in practice only atoms within a cutoff radius of atom i are necessary to construct each feature vector. (Here, an "image" is defined as a single geometric configuration $\vec{\mathbf{R}}$.) A machine-learning model is fit at the per-atom level to give $\hat{E}_i$ as a function of the feature vector; this model is typically specific to each atomic element. Finally, the potential energy prediction is found by summing the per-atom contributions. Since the force on each atom is the negative gradient of the potential energy with respect to that atomic position,

$$\vec{\mathbf{F}}_i = -\vec{\nabla}_i E, \tag{2}$$

the predicted forces can be found by analytical or numerical differentiation of the predicted energy with respect to the atomic positions $\vec{\mathbf{R}}$. The feature vectors can be produced by numerous transformations (Gaussian functions,[4] Zernike descriptors,[7] bispectrums,[5] etc.) and the machine-learning can consist of various regression models (neural networks,[4] Gaussian processes,[5,12] kernel ridge regression,[6] etc.). Gaussian feature vectors employed in this work are described in the Supporting Information (SI). In practice, a set of training images is produced with an electronic-structure calculator, *e.g.*, DFT. These are transformed to produce feature vectors, and a machine-learning model is regressed to match the energies and forces to within a user-specified tolerance. We have released an open-source software package, known as *Amp* (Atomistic Machine-learning Package),[7,13] which allows the user to create such models while independently specifying feature vector transformations and learning models.

Since all machine-learning models are "black-box" approaches which only emulate another functional form by learning from training data, we can expect that the predictions of the model may poorly represent the true function in regions where training data is inadequate. However, the identification of such regions is challenging in the $3N$-dimensional space in which atomic simulations take place, unless self-validating algorithms are used to verify key points with the parent calculator.[14] We suggest that atomistic machine-learning models will only see widespread use when they can be demonstrated to be trusted; thus, it is essential to understand and develop methods to address uncertainty in these calculations.

We can divide the reasons that a model may give a poor prediction for the potential energy (or forces) for an atomic configuration $\mathbf{R}$ into two categories:

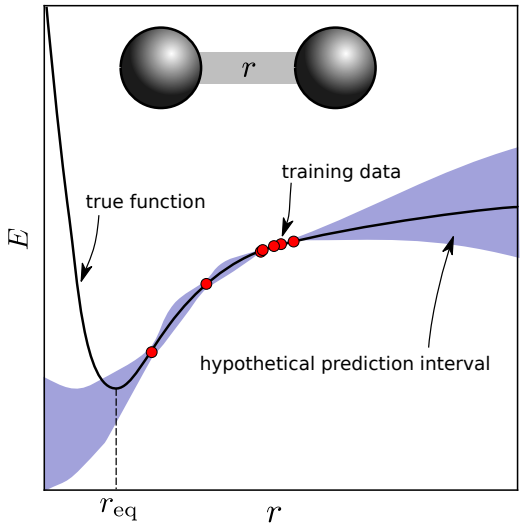

Figure 1: Potential pitfalls in estimating uncertainty in atomistic machine learning. The functional forms of the potential energy surface often encountered in atomistic simulations are likely to cause unreliable prediction intervals for black-box methods if the training data doesn't adequately sample the space. This suggests that prediction intervals should not be interpreted as quantitative predictions of uncertainty, but should be taken only as an indication that new *ab initio* data is necessary when an increased prediction interval is encountered. Shown is a classic interatomic potential for a covalently bound system, in which all of the training data were taken when $r > r_{\mathrm{eq}}$; we would expect any machine learning model exposed to these data to give a poor representation due to the steep change in slope as $r < r_{\mathrm{eq}}$.

- **Type 1:** Insufficient data. That is, inadequate training data is present for the model to have learned how to calculate E in the region of $\mathbf{R}$.
- **Type 2:** Incapable model. The functional form of the model is incapable of describing E at $\mathbf{R}$. This can occur for many reasons, such as the coordinate transformation used, the type or size of the machine-learning regression model, or the presence of non-local phenomena in the electronic-structure calculations, such as an external field or long-range electronic interactions.

In this work, we are primarily focused on the first type, but we briefly discuss Type-2 errors here. First, consider the case where adequate training data is present: if a model is incapable of fitting the data (a Type-2 error) it will be obvious, as there will be large errors on the training set; that is, the model will not be able to reproduce the training data. This can be addressed by a number of standard means, such as increasing the cutoff radius, the model parameter space, or adding in corrections for non-local effects. Next, consider if a (converged) model is asked to make a prediction in a region of the PES in which it is mathematically incapable of fitting the parent function: this is necessarily in a region of the PES where we do not have sufficient training data, as otherwise the model would not have converged. Therefore, properly addressing Type-1 errors should greatly reduce the likelihood that Type-2 errors appear.

We also note that atomistic calculations provide some unique challenges for machine learning. First, since the general purpose of atomistic simulations is to explore (new structures, new proper-

ties, new reactions, etc.), we should expect the user to frequently move into regions of the PES in which training examples are sparse or non-existent. This is fundamentally different than many applications of machine learning, in which future scenarios can be considered to be well-represented by a large set of past behavior. Thus, it is not sufficient to assess a model's fit by dividing the example data into training and validation sets, as the validation set no longer provides a good measure of the goodness-of-fit for the new region. Further, the transformation of the coordinates into feature space can make it difficult to understand when a user has moved into an unexplored region of the PES. Second, we argue that the functional forms encountered in atomistic simulations may make uncertainty bounds on predictions unreliable. Consider the classic functional form taken by the potential energy as the distance between two covalently bound atoms is varied, as shown in Figure 1. If the training data is sampled only when $r > r_{\mathrm{eq}}$ (where r_{eq} corresponds to the minimum energy bond length), we would not expect any "black-box" regression model to correctly predict the steep rise in energy as r becomes less than r_{eq}. Of course, one could circumvent the "black box" approach of machine learning and anticipate such functional forms; however, more complicated structural rearrangements, such as torsional angles, can also lead to similar phenomena, and the explicit anticipation of such effects is best left to traditional force-field development. For these reasons, we would not expect to be able to rigorously trust uncertainty bounds when dealing with atomistic data that doesn't adequately sample the PES.

However, we anticipate uncertainty bounds to still be very useful in such simulations. As we will discuss in upcoming examples, if the uncertainty bounds on a new calculation (significantly) exceed the uncertainty bounds on the training data, it suggests that we have left the region of the PES where the training data was created, and therefore new training data should be created from this atomic configuration. That is, an electronic-structure calculation should be undertaken and the machine-learning calculator re-trained with the inclusion of this point, thus improving the model in this region. Note that this will only happen when the prediction interval is large, so a new calculation is justifiable even if the uncertainty bounds did provide quantitative uncertainty estimates. On the other hand, if uncertainty bounds are within the range of those of the training data, the uncertainty bounds should be capable of providing a reliable estimate of the anticipated error in the machine-learning estimate.

In the examples in this Perspective, we will generate uncertainty ranges through the creation of an ensemble of machine-learning calculators, produced with a bootstrap[15] resampling technique. Other approaches to dealing with uncertainty in atomistic machine learning have been suggested. For example, Behler[11] proposed the use of a committee of several trained calculators as a means to systematically expand the training set in relevant regions of configurational space for a given problem. In his suggestion, the neural networks should have different functional forms (such as the structure of the hidden layers) in order to introduce independence into the models. This committee of two or more models would then be used to perform molecular dynamics simulations; when images are encountered for which the committee's energy predictions disagree, those images would be added to the training set to improve the calculator. Elsewhere, Li *et al.*[16] used Gaussian processes as a force-predictor in molecular dynamics simulations; since Gaussian processes can give an inherent indication of uncertainty, this enabled them to predict when new quantum-mechanics based information was necessary in a predictor–corrector scheme.

In the bootstrap approach we implement here, each calculator of the ensemble is trained to a different, randomly-selected set of training images. Specifically, the images are drawn from the parent set of N training images, randomly with replacement, until a new training set of N

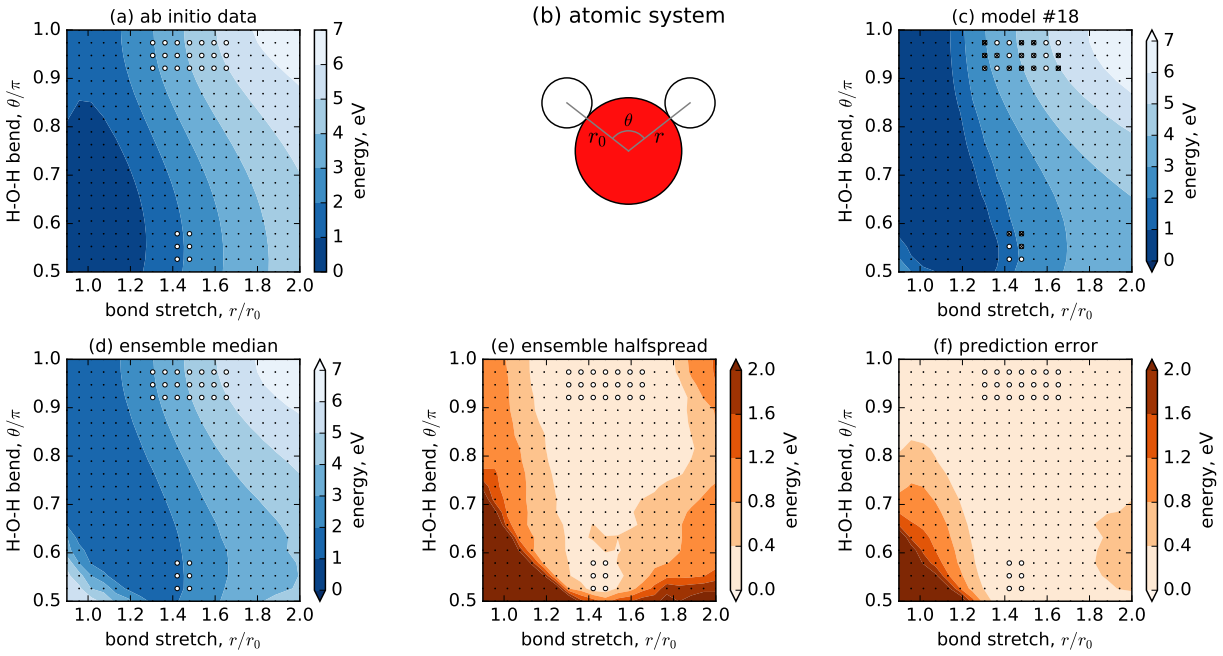

Figure 2: Example of addressing uncertainty on a simple PES describing a water molecule. (a) The potential energy surface as calculated with DFT. The points are the individual DFT calculations and the shading represents the potential energy; the open circles are points to be employed as training data. (b) The coordinates of the water molecule: r_0 is fixed while r and θ are variable. (c) A single calculator from the bootstrap ensemble. Here the open circles represent the available training data while those filled with crosses are those points randomly chosen for this single calculator. (d) The median prediction of an ensemble of 50 calculators. (e) The "halfspread" of the ensemble, defined as the spread between the 5th and 95th percentiles of the ensemble divided by two. (f) The prediction error of the ensemble, defined at each point as the absolute difference between the median ensemble prediction and the DFT data.

images is created. This is repeated for the training set of each calculator in the ensemble. On average, each image appears at least once in about 63% of the training sets chosen, in the limit of large training sets. (For comparison, if an ensemble were created with a more conventional k-fold cross-validation technique, each calculator in the k-member ensemble would contain $\frac{k-1}{k}$ ($\times 100\%$) of the original training points, making them nearly identical and highly correlated, approaching the behavior of a committee machine as the ensemble size grows. In the bootstrap resampling technique, the sampling probability remains constant, allowing the ensemble's behavior to converge as the ensemble size grows.) Importantly, the bootstrap resampling approach forces each calculator to have an independent training set, and places greater statistical certainty on regions of the PES where training data is more dense, while leading to a greater disagreement between models where data is relatively sparse.

Example systems. We created two model systems for this Perspective which allow us to examine uncertainty in well-defined atomic systems. To create a system in which the PES can be entirely visualized in two dimensions, we produced simulations of a single water molecule in which one O–H bond length and the H–O–H angle are varied, keeping the other O–H bond at its equilibrium

position from the relaxed molecule. We created a grid of DFT data (using the NWChem calculator[17] with the B3LYP functional[18,19] and the 6-31+G* basis set,[20] as described in the SI). The grid of DFT data points is shown along with the DFT potential energy surface in Figure 2(a) while the atomic system that describes the PES coordinates is shown in Figure 2(b). For the current example, we chose the 27 points marked on the figure with open circles as the available training data, and we will assess the ability to fit other regions of the PES marked with solid dots.

We used a bootstrap resampling technique[15] to create a 50-member ensemble of machine-learning calculators to approximate the potential energy surface; that is, each calculator's training set was chosen from the original 27-image set, with replacement, to form a new 27-image set. Each machine-learning calculator in the ensemble was created in *Amp*[7,13] using a Behler–Parrinello scheme[4] (that is, with Gaussian feature vectors and neural network regression models), but the discussion is by no means limited to such a model choice. For this initial discussion, we regressed the model to fit the potential energies of the training images, and ignored the force information produced by the DFT calculations. Later, we will discuss the effect of including forces during the regression. Full details are provided in the SI; we have made both *Amp* and the bootstrap technique we employed available as open-source software.[13]

The PES predicted with a single calculator of the ensemble is shown in Figure 2(c). Here, the chosen training images are marked with a cross on the PES. As required by the convergence criterion when fitting the model, we see a near-perfect fit in the regions of these chosen points, which can be observed by comparing to subfigure (a). We can see that the model does a reasonable job of predicting the PES between the two regions of sampled points, and the model's estimate deviates more significantly from the true function in regions without training data.

By forming an ensemble of such calculators, a median prediction as well as a prediction interval of the ensemble can be determined. The median ensemble prediction is shown in Figure 2(d). Again, we see excellent agreement in the regions with training data, and poorer predictions as we move away from this data. The halfspread of the data—defined as one-half of the spread between the 5th and 95th percentiles of the ensemble prediction—is shown in Figure 2(e). (Note the halfspread, rather than the spread, is used to allow comparison to residuals.) Correctly, we see a small halfspread in regions where training data is present as well as in regions where the ensemble gives good predictions, which can be seen by comparison to subfigure (f). Farther away from the training data, the halfspread increases as the calculators in the ensemble start to diverge from one another, and in all regions where there is a large error in prediction, there is also a large halfspread predicting (the possibility of) this error.

Of course, practical atomistic calculations will have many more degrees of freedom and cannot be visualized in a two-dimensional PES. In Figure 3(a), we collect the data from Figures 2(e) and (f), plotting the prediction residuals versus the ensemble halfspread. The training points are shown as red circles while the rest of the points (marked "test") are shown as black dots. The halfspread on the training data can be seen to be relatively low in all cases, and the difference in halfspread among the training examples can be traced to the density of training points in the neighborhood of each point. For the points not included in the training set, if the halfspread is similar to that of the training data, then it is reasonable to expect that statistically meaningful prediction intervals can be crafted (see Heskes[15] for an example procedure); for the present discussion, we simply compare to the parity line, and see that the data fall below the parity line, in the shaded region. Here we see that this is still generally the case even at larger halfspreads, with some exceptions when the halfspreads grow large.

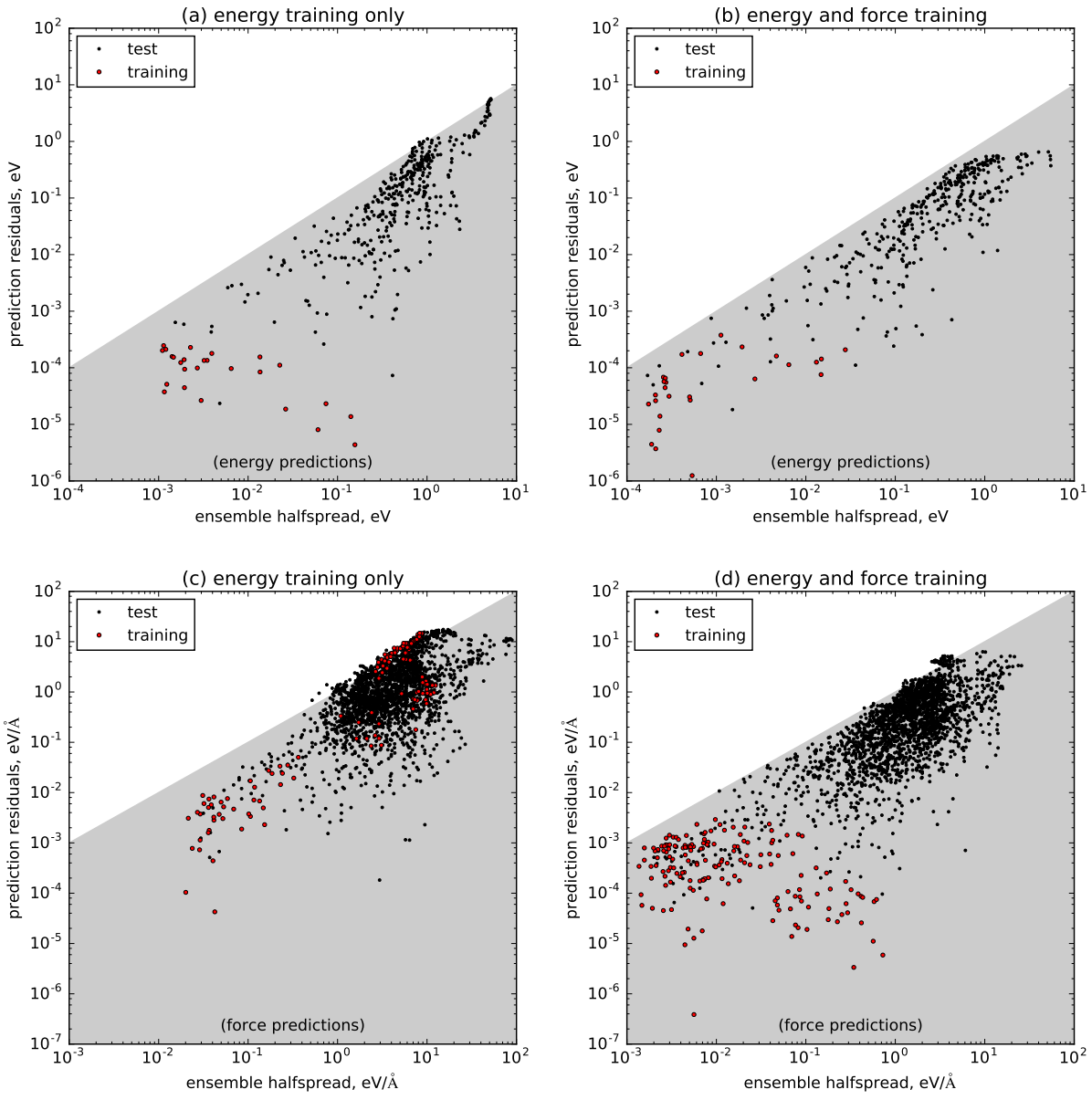

Figure 3: Prediction accuracy versus ensemble spreads. (a) and (b) show predictions and residuals for potential energies, while (c) and (d) show predictions for forces. In the case of (a) and (c), only the potential energy was regressed to fit the ensemble; in (b) and (d) both the potential energy and forces were regressed.

Most electronic-structure calculators provide not just the energy, but also the forces, and if the machine-learning model is also fit to the forces (by including them in the loss function), then the machine-learned representation of the PES should improve relative to including only the energies.[7,21] To compare this, we trained a second bootstrap ensemble of 50 calculators to both energies and forces, for the same 27 images. In Figure 3(b), we see this caused a general improvement to the ensemble's estimate of the energies, even though the individual calculators were converged to the same energy tolerance (discussed in the SI). This is to be expected, as the inclusion of the gradient provides much richer information and therefore much tighter constraints on the form of the PES. Interestingly, we also see a reduction in the halfspread of the training images. This can be understood by considering that the variation in the ensemble prediction in the region of training images is introduced primarily by randomly omitting individual points; however, if a point is omitted from an individual calculator, the inclusion of the gradient at a neighboring point leads the calculator to make better predictions—and thus a reduced ensemble halfspread—even at an omitted point. Also, we see that the general concept holds in both cases—the data is primarily below the parity line.

We also examine the uncertainty in force predictions; in Figure 3(c-d) we use an identical technique on the force predictions for each atom in the simulation. Unsurprisingly, when forces are not included in the regression technique (subfigure (c)), the prediction of forces is relatively poor, even for much of the training data. When forces are included in the regression, we see a lower spread in general and a significantly lower spread for the training points. In general, we can see that the prediction intervals work quite well in all these cases.

To examine a second system with more degrees of freedom, we created a Pt fcc (111) surface and propagated Langevin dynamics (as implemented in ASE[22]) for 10,000 5-fs steps. Importantly, this trajectory included the formation of a vacancy; that is, a surface atom moved from a close-packed layer to an adatom position. The atomic system is shown in Figure S1 of the SI. We can see the creation of the vacancy by tracing the z coordinate of the vacancy-creating atom, shown in Figure 4(a) and occurring around step 7,000. The electronic-structure calculations were performed in planewave/pseudopotential DFT with the Dacapo calculator[22] and the RPBE functional;[23] full details of the trajectory creation and the electronic-structure calculations are provided in the SI.

We randomly sampled 100 images from the vacancy-free time interval before step 6,000 and created a 50-member bootstrap ensemble from these 100 training images, using the same machine-learning scheme as in the previous example. By creating the ensemble from only vacancy-free images, this facilitated the examination of the ensemble behavior both in established regions of the PES (before vacancy creation) and as a new region of the PES is entered (upon vacancy creation). Figure 4(b) shows the potential energy trace for the "true" (DFT) data overlayed with the median ensemble prediction. In the time period before the defect is created, we see the ensemble prediction largely traces the DFT-calculated data, but the two diverge much more frequently and dramatically after the vacancy is created. The ensemble halfspread is shown in Figure 4(c). Encouragingly, we see a relatively low halfspread in the time period before the defect is created, and a large halfspread after it is created. The large halfspread in the defect time period corresponds well to the poor fit in this region. Interestingly, we see the halfspread begin to increase well before the atom changes significantly in height; presumably, this corresponds to a higher system disorder as the defect is beginning to form; nevertheless, the ensemble predicts well the divergence between the median-predicted and DFT-calculated energies in this region. Note also that the ensemble halfspread also correctly predicts inaccuracies at around step 600 and again at around step 5,500, both in the vacancy-free region.

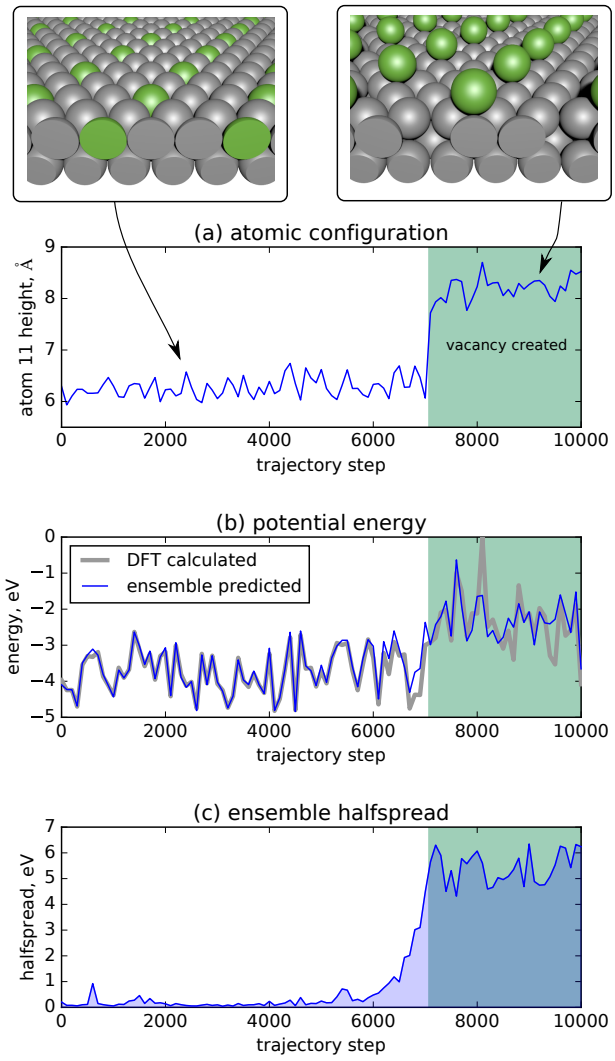

Figure 4: Example of defect creation on an fcc (111) Pt surface. In each trace, every 100th image is shown for clarity. (a) z position of atom #11 which creates the defect. The z direction is normal to the surface. (b) Trace of DFT-calculated and ensemble-predicted potential energy. The reference energy is the maximum energy of the DFT data set. (c) Trace of ensemble halfspread.

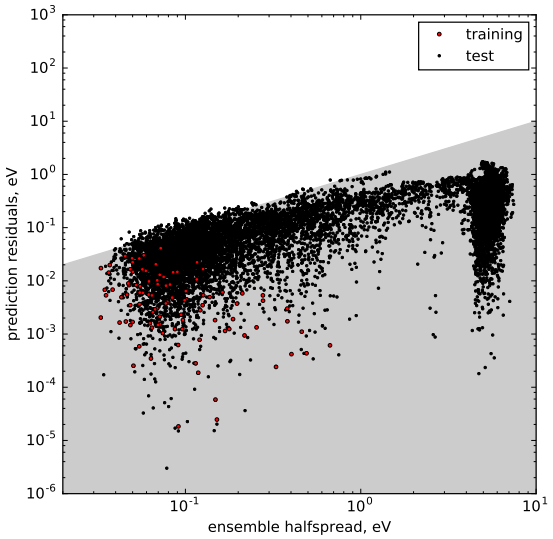

Figure 5: Ensemble halfspread versus absolute residuals for the defect creation example.

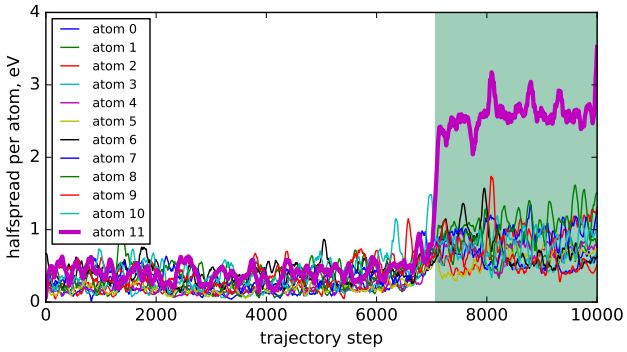

Figure 6: Localized ensemble spread for the defect creation example. To reduce noise in the time sequence, a Savitzky–Golay filter is applied to the data before plotting.

The results in the trajectory trace in Figure 4 show only a small fraction of the data. To examine the full data set, we use the same analysis tool as in the previous example, shown in Figure 5. The results for this much larger data set of 10,000 images show similar findings as in the simple H_2O example. Again, we see that the ensemble halfspread is a very good indicator of the prediction residuals for both the training and test sets, and in this case is even a conservative predictor. However, for the reasons discussed above, we again suggest that the best interpretation of a wide halfspread is not as a prediction of the uncertainty range, but as an indication that if the halfspread is larger than that of the training data points, then additional *ab initio* calculations should be performed in the region of interest. In the SI, we outline a scheme for minimizing the computational expenses of re-training an existing ensemble when images are added to the training data.

A surprisingly useful feature of the atom-centered approach is that the uncertainty can be localized in the simulation. Due to the separability of energy in equation (1), an ensemble halfspread can be created for each value of E_i; that is, the error can be isolated per atom. For our vacancy-creation example, the per-atom spread is shown in Figure 6. We see that in the region of the defect

creation (shaded region), the ensemble halfspread of each atom increases to some extent. However, the strongest contributor to the error can be isolated to atom #11, which is the atom that moves onto the surface to create the defect. This identifies the atom which has the local environment most different from the local environments present in the training set. In large and complicated simulations, we expect this approach to be useful to isolate specifically what geometric changes caused the uncertainty. This even suggests that in large images with a high uncertainty, partial systems can be constructed (and calculated with the parent electronic-structure calculator) in order to systematically improve the error.

Outlook. Machine-learning regression of the potential energy surface has provided many impressive demonstrations of emulating higher-accuracy calculations. However, these techniques have seen limited application and, arguably, have not yet provided significant new atomistic insights or accelerations to discovery. We contend that this is due to an inability to trust the results on unseen systems, which highlights the urgent need for uncertainty analysis in machine learning.

In the current work, we have highlighted the usefulness of uncertainty analysis in atomistic machine learning, and demonstrated the utility with a method that was shown to be powerful in discovering regions of (un)certainty in these examples. Of course, many other methods of addressing uncertainty are possible, and the usefulness of each will depend upon the specifics of the application. The uncertainty analysis typically comes with a cost; in our case of a 50-member ensemble, this means that the energy (and/or forces) must be calculated fifty times per energy (and/or force) call. Fortunately, calculation of the feature vector and its derivative is identical for each calculator in the ensemble, so the true computational cost can be considerably less than $50\times$, depending upon the computational demands of the feature-vector creation versus the machine-learning portion of the calculation. Similar arguments hold for training of the ensemble; in the SI, we propose a scheme for the efficient re-training of an ensemble when new image(s) are added to the training examples. Since machine-learning calculators are typically several orders of magnitude faster than electronic-structure calculations, the cost of employing an ensemble is relatively minor in order to add a level of trust to calculations.

While we showed bootstrap resampling to be an easily implementable approach to address uncertainty in atomistic machine learning, we expect that new and interesting approaches will be adopted in the future to supply more statistical rigor at greater computational efficiency. Many approaches are possible, such as the calculation of the similarity of feature vectors of new examples to those in the training set,[24] the use of a committee machine,[11] or the use of methods such as Gaussian processes with more built-in uncertainty. As an alternative to an ensemble of independently trained calculators, an ensemble could be formed from a single trained calculator via the perturbation of parameters with Bayesian methods.[25] Google's DeepMind project recently suggested[26] an algorithm to treat each parameter as a distribution during training rather than a numerical value. With this computationally efficient algorithm ("Bayes-by-Backprop") an in-principle infinitely large ensemble of calculators is trained simultaneously, under the assumption that all parameters are normally distributed about optimum values. Since only a single loss function minimization needs to be performed, this method can significantly reduce training cost, and the energy-/force-call cost would approach the limit of $2\times$, as each parameter carries a mean and variance. It is not clear, however, if ensembles formed in these manners will provide as trustworthy an estimate as the bootstrap method,[25] since an implicit assumption is made that model variability can be captured in the variance about a local minimum in the loss function. Logically, this is more likely to result in

"false negatives" if the ensemble calculators collectively fail to reproduce the electronic-structure calculations.

If procedures for uncertainty analysis can be coupled with atomistic machine learning, we expect that machine learning will become much more widely adopted in the atomistics community, and that new applications will arise. As an example, we can envision a database-enabled calculator that provides electronic-structure calculations "at the speed of search". In this vision, a centralized calculator builds upon the calculations of thousands of researchers: a user can submit a calculation, and the centralized calculator nearly instantly provides the machine-learned estimate of the energy/forces, as well as an indication of the trustworthiness of the result. If the calculation has a high degree of certainty, the user accepts the results and moves on to the next calculation. If not, the user performs a new *ab initio* calculation, automatically submitting the result to the centralized computer's database, which subsequently improves the model for all users. In such a scheme, the model would continuously improve due to the computational efforts of many users, and would improve strategically in regions of the greatest need. We expect many such new approaches to emerge if we can provide trust to the predictions of atomistic machine learning.

ASSOCIATED CONTENT

Supporting information. The supporting information contains background information on the bootstrap resampling procedure employed, a strategy to grow a bootstrap ensemble as new training images become available, and calculation details for the two examples discussed in the manuscript.

Acknowledgments. The authors gratefully acknowledge insightful discussions with and/or inspirations from Nongnuch Artrith (MIT), Zack Ulissi (Stanford), Tejs Vegge (DTU), and Brandon Wood (Lawrence Livermore National Laboratory). Calculations were performed at Brown University's Center for Computation and Visualization.

References

[1] Cramer, C.J. *Essentials of Computational Chemistry*. Wiley, 2nd edition **2004**.

[2] Blank, T.B.; Brown, S.D.; Calhoun, A.W.; Doren, D.J. Neural network models of potential energy surfaces. *The Journal of Chemical Physics* **1995**; 103, 4129–4137.

[3] Lorenz, S.; Groß, A.; Scheffler, M. Representing high-dimensional potential-energy surfaces for reactions at surfaces by neural networks. *Chemical Physics Letters* **2004**; 395, 210–215.

[4] Behler, J.; Parrinello, M. Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Phys. Rev. Lett.* **2007**; 98, 146401.

[5] Bartók, A.P.; Payne, M.C.; Kondor, R.; Csányi, G. Gaussian Approximation Potentials: The Accuracy of Quantum Mechanics, without the Electrons. *Phys. Rev. Lett.* **2010**; 104, 136403.

[6] Botu, V.; Ramprasad, R. Learning scheme to predict atomic forces and accelerate materials simulations. *Phys. Rev. B* **2015**; 92, 094306.

[7] Khorshidi, A.; Peterson, A.A. *Amp*: A modular approach to machine learning in atomistic simulations. *Computer Physics Communications* **2016**; 207, 310 – 324.

[8] Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Networks* **1989**; 2, 359–366.

[9] Behler, J. Perspective: Machine learning potentials for atomistic simulations. *The Journal of Chemical Physics* **2016**; 145, 170901.

[10] Handley, C.M.; Popelier, P.L.A. Potential Energy Surfaces Fitted by Artificial Neural Networks. *The Journal of Physical Chemistry A* **2010**; 114, 3371–3383.

[11] Behler, J. Representing potential energy surfaces by high-dimensional neural network potentials. *Journal of Physics: Condensed Matter* **2014**; 26, 183001.

[12] Koistinen, O.P.; Maras, E.; Vehtari, A.; Jónsson, H. Minimum energy path calculations with Gaussian process regression. *Nanosystems: Physics, Chemistry, Mathematics* **2016**; 7, 925–935.

[13] Khorshidi, A.; Ulissi, Z.; El Khatib, M.; Peterson, A. Amp: The Atomistic Machine-learning Package v0.5. *Zenodo* **2017**; DOI:10.5281/zenodo.322427.

[14] Peterson, A.A. Acceleration of saddle-point searches with machine learning. *The Journal of Chemical Physics* **2016**; 145, 074106.

[15] Heskes, T. Practical confidence and prediction intervals. In M.C. Mozer; M.I. Jordan; T. Petsche, eds., *Advances in Neural Information Processing Systems 9*. MIT Press. 1996.

[16] Li, Z.; Kermode, J.R.; De Vita, A. Molecular Dynamics with On-the-Fly Machine Learning of Quantum-Mechanical Forces. *Phys. Rev. Lett.* **2015**; 114, 096405.

[17] Valiev, M.; Bylaska, E.; Govind, N.; Kowalski, K.; Straatsma, T.; Dam, H.V.; Wang, D.; Nieplocha, J.; Apra, E.; Windus, T.; de Jong, W. NWChem: A comprehensive and scalable open-source solution for large scale molecular simulations. *Computer Physics Communications* **2010**; 181, 1477 – 1489.

[18] Becke, A.D. Density-functional exchange-energy approximation with correct asymptotic behavior. *Phys. Rev. A* **1988**; 38, 3098–3100.

[19] Lee, C.; Yang, W.; Parr, R.G. Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density. *Phys. Rev. B* **1988**; 37, 785–789.

[20] Ditchfield, R.; Hehre, W.J.; Pople, J.A. Self-Consistent Molecular-Orbital Methods. IX. An Extended Gaussian-Type Basis for Molecular-Orbital Studies of Organic Molecules. *The Journal of Chemical Physics* **1971**; 54, 724–728.

[21] Pukrittayakamee, A.; Malshe, M.; Hagan, M.; Raff, L.M.; Narulkar, R.; Bukkapatnum, S.; Komanduri, R. Simultaneous fitting of a potential-energy surface and its corresponding force fields using feedforward neural networks. *The Journal of Chemical Physics* **2009**; 130, 134101.

[22] Bahn, S.R.; Jacobsen, K.W. An object-oriented scripting interface to a legacy electronic structure code. *Computing in Science & Engineering* **2002**; 4, 56–66.

[23] Hammer, B.; Hansen, L.B.; Nørskov, J.K. Improved adsorption energetics within density-functional theory using revised Perdew-Burke-Ernzerhof functionals. *Physical Review B* **1999**; 59, 7413–7421.

[24] Botu, V.; Ramprasad, R. Adaptive machine learning framework to accelerate ab initio molecular dynamics. *International Journal of Quantum Chemistry* **2015**; 115, 1074–1083.

[25] Christensen, R. *Error Mitigation in Computational Design of Sustainable Energy Materials*. Ph.D. thesis, DTU Energy, Technical University of Denmark **2016**.

[26] Blundell, C.; Cornebise, J.; Kavukcuoglu, K.; Wierstra, D. Weight Uncertainty in Neural Networks. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15. JMLR.org, pp. 1613–1622.